

J.L.S. INFORMATIQUE

2, rue Clément ADER

B.P. 50065

57972 YUTZ CEDEX

Tel : 33 (0)3 82 86 00 16

Fax : 33 (0)3 82 86 00 12

URL : www.jls-info.com

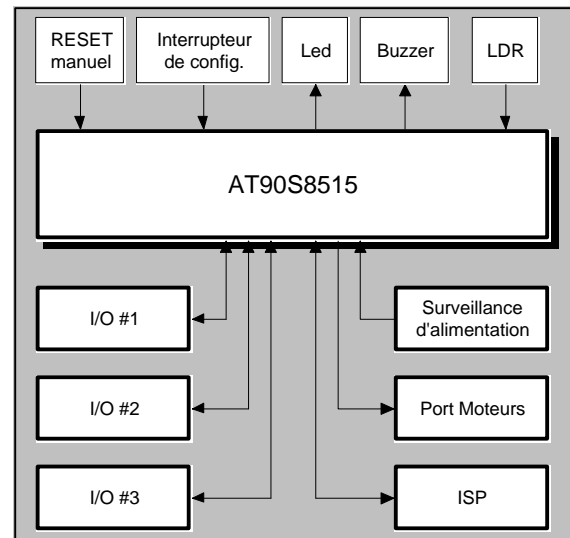
MCU8515

Carte universelle à AT90S8515

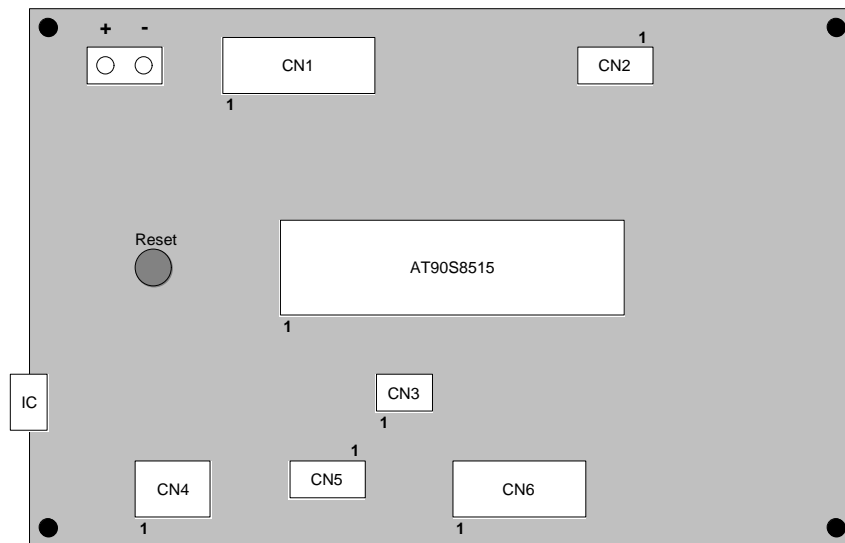
Caractéristiques générales

La carte **MCU8515** est une carte universelle à microcontrôleur 8 bits équipée de 8Ko de FLASH. La Elle est au format 103mm x 65mm et dispose de nombreuses ressources :

- Microcontrôleur AT90S8515 à 8MHz
- 8Ko de FLASH interne, fonction ISP
- 512o de RAM statique interne
- 512o d'EEPROM interne
- UART TTL
- 24 I/O partagées (*Timer, UART, Interrupt...*)
- Connecteur pour contrôle de moteurs DC
- Surveillance d'alimentation
- Dimensions 103mm x 65mm
- Fonctionnement de 0°C à 60°C
- Interrupteur de configuration, Led bicolore, buzzer
- Alimentation > 7.5V, consommation maximum de 40mA



Brochage des connecteurs



Vue du dessus de la carte

CN1 : I/O #1

PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
GND	GND	GND	GND	GND	GND	GND	GND
VCC	VCC	VCC	VCC	VCC	VCC	VCC	VCC
1	2	3	4	5	6	7	8

CN2 : Port moteurs

9	7	5	3	1
GND	GND	PC2 / EN	PC1 / SMD	PC0 / SMG
GND	GND	GND	GND	GND
10	8	6	4	2

SMD : Sens Moteur Droit
 SMG : Sens Moteur Gauche
 EN : Enable

Ces trois signaux sont affectés au pilotage de deux moteurs DC, par l'intermédiaire d'une carte de puissance (Article paru dans la revue *Micros Et Robots* n°1). Si cette fonctionnalité n'est pas utilisée, les I/O PC0, PC1 et PC2 sont disponibles comme ports normaux.

CN3 : Port UART TTL

VCC	PD0 / RD	PD1 / TD	GND
1	2	3	4

CN4 : I/O #3

PB0 / T0	PB1 / T1	PB2 / Ain0	PB3 / Ain1
GND	GND	GND	GND
VCC	VCC	VCC	VCC
1	2	3	4

CN5 : Port ISP

9	7	5	3	1
GND	/RESET	PB5 / MOSI	PB6 / MISO	PB7 / SCK
GND	GND	GND	GND	GND
10	8	6	4	2

CN5 est une interface de programmation ISP. Cela permet au programmeur de tester ses programmes un à la suite de l'autre sans mettre hors tension le module et sans sortir le composant pour le placer sur un programmeur adéquat. La liaison se fait au travers du câble fourni HE10 / DB25. Branché sur le port parallèle du PC et avec un logiciel fort simple*, le microcontrôleur se place en Reset, puis en programmation (Flash, EEPROM, et vérification si nécessaire) puis branche en début de code utilisateur.

Si cette fonctionnalité n'est pas utilisée, les I/O PB5, PB6 et PB7 sont disponibles comme ports normaux. La ligne /RESET, est directement reliée à la broche /RESET du microcontrôleur et au bouton – poussoir RESET.

CN6 : I/O #2

OC1B	PD2 / INT0	PD3 / INT1	PD4	PD5 / OC1A	PD6 / WR	PD7 / RD
GND	GND	GND	GND	GND	GND	GND
VCC	VCC	VCC	VCC	VCC	VCC	VCC
1	2	3	4	5	6	7

Connexion des I/O du microcontrôleur

	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
Port A	std	std	std	std	std	std	std	std
Port B	isp / std	isp / std	isp / std	IC	std	std	std	std
Port C	SA	LV	LR	BZ	LDR	EN / std	SMD / std	SMG / std
Port D	std	std	std	std	std	std	TD / std	RD / std

std : I/O standard, éventuellement partagée avec une ressource interne

isp : liaison ISP

IC : Interrupteur de configuration Run (1) / Test (0) ; *read only*

SA : surveillance d'alimentation ; *read only*

LV, LR : Led Verte, Led rouge ; *write only*

BZ : buzzer ; *write only*

LDR : LDR ; *read only*

EN, SMD, SMG : commandes moteur DC (*Enable*, Sens Moteur Droit, Sens Moteur Gauche)

TD, RD : lignes UART

read only : il est impératif de configurer ces I/O en entrée (registre DDRx correspondant)*

write only : il est impératif de configurer ces I/O en sortie (registre DDRx correspondant)*

* : voir codes sources concernés

Codes sources

Utilisation de la Led bicolore

```
; equivalences
.equ  ledv  = pc6
.equ  ledr  = pc5

...
        ldi    temp,$60           ;initialisation
        out    ddrc,temp         ;pc5 et pc6 en sortie
        cbi    portc,ledv        ;led v off
        cbi    portc,ledr        ;led r off

...
        sbi    portc,ledv        ;utilisation #1
        cbi    portc,ledv        ;led v on
        cbi    portc,ledv        ;led v off
        sbi    portc,ledr        ;led r on
        cbi    portc,ledr        ;led r off

...
        sbi    portc,ledv        ;utilisation #2
        sbi    portc,ledr        ;led orange on

...
```

Utilisation du buzzer

```
; equivalences
.equ  bz    = pc4
.equ  cpt   = r17                ;compteur

...
        ldi    temp,$10         ;initialisation
        out    ddrc,temp        ;pc4 en sortie

...
        ldi    cpt,50           ;utilisation
                                ;init compteur
sound:
        sbi    portc,bz         ;front montant
        rcall  delay            ;tempo
        cbi    portc,bz         ;front descendant
        rcall  delay            ;tempo
        dec    cpt
        brne   sound           ;50x ?

...
delay:
        ldi    tempo,$ff        ;boucle executee 256 fois, fixe la frequence du son
delayy:
        dec    tempo
        brne   delayy          ;tempo = 0 ?
        ret

...
```

Utilisation de l'interrupteur de configuration

```
.equ   ledv    = pc6           ;led v
.equ   ic      = pb4           ;interrupteur de configuration

...
                                ;initialisation
    ldi    temp,$00
    out    ddrb,temp           ;portb en entree
    ldi    temp,$40
    out    ddrc,temp          ;pc6 en sortie

...
                                ;utilisation
    sbis   pinb,ic             ;teste valeur ic
    rjmp   led_v_on            ;branche si 0
    rjmp   led_v_off           ;branche si 1
led_v_on:
    sbi    portc,ledv          ;led v on
    rjmp   main
led_v_off:
    cbi    portc,ledv          ;led v off

...
```

Utilisation de la LDR

```
.include "8515def.inc"

.def   temp    = r16           ;mcu
.equ   ledr    = pc5           ;led rouge
.equ   ldr     = pc3           ;ldr

...
                                ;initialisation
    ldi    temp,$70
    out    ddrc,temp           ;pc4, pc5 et pc6 en sortie
    ldi    temp,$10
    out    portc,temp          ;buzzer, led v et led r off

...
                                ;utilisation
main:
    sbis   pinc,ldr            ;branche si eclairage ok
    rjmp   led_r_off
led_r_on:
    sbi    portc,ledr
    rjmp   main
led_r_off:
    cbi    portc,ledr
    rjmp   main
```

Surveillance de la tension d'alimentation

L'alimentation demandée par la carte est une tension continue, supérieure à 7.5V. Un circuit de surveillance a été mis en place, dans le cadre d'une application embarquée fonctionnant sur batterie.

Seuils de détection :

- Valim > 7.4V : bit SA = 0
- Valim < 7.4V : bit SA = 1

Cet exemple de programme génère une interruption temps réel toutes les 500ms, et émet un son bref. Dans l'attente de cette interruption, le programme principal teste l'état du bit SA et allume la led rouge si la tension d'alimentation descend en dessous de 7.4V

```
.include "8515def.inc"

.def    temp    = r16        ;mcu
.def    tempo   = r17        ;tempo
.def    cpt     = r18        ;compteur pour bz

.equ    sa      = pc7        ;surveillance d'alimentation
.equ    ledv    = pc6        ;led verte
.equ    ledr    = pc5        ;led rouge
.equ    bz      = pc4        ;buzzer
.equ    ic      = pb4        ;interrupteur de config

.cseg
.org    $0000
rjmp   reset          ;reset
rjmp   no_isr         ;ext int 0
rjmp   no_isr         ;ext int 1
rjmp   no_isr         ;timer1 capture
rjmp   no_isr         ;timer1 compare a
rjmp   no_isr         ;timer1 compare b
rjmp   tl_isr        ;timer1 overflow
rjmp   no_isr         ;timer0 overflow
rjmp   no_isr         ;spi transfert complete
rjmp   no_isr         ;rx uart
rjmp   no_isr         ;udre uart
rjmp   no_isr         ;tx uart
rjmp   no_isr         ;comparator
no_isr:
    reti

init_syst:
    ldi    temp,$70
    out   ddrc,temp    ;pc4, pc5 et pc6 en sortie
    cbi   portc,ledv
    cbi   portc,ledr    ;leds v et r off
    cbi   portc,bz      ;buzzer repos
    ldi   temp,$f0      ;real time interrupt
    out   tcntlh,temp   ;65535-(rti_time*ck/1024)
    ldi   temp,$bd      ;rti_time=500ms
    out   tcntll,temp
    ldi   temp,$05
    out   tcctlb,temp   ;init timer1 : ck/1024
    ldi   temp,$80
    out   tmsk,temp
    sei
    ret

delay:
    ldi   tempo,$ff
delayy:
    dec   tempo
    brne delayy
    ret
```

```

t1_isr:
    ldi    temp,$f0
    out   tcnt1h,temp
    ldi    temp,$bd
    out   tcnt1l,temp    ;rti_time=500ms
    ldi    cpt,50
sound:
    sbi    portc,bz
    rcall  delay
    cbi    portc,bz
    rcall  delay        ;signal sur buzzer
    dec   cpt
    brne  sound        ;boucle 50x ?
    reti

reset:
    ldi    temp,high(ramend)
    out   sph,temp
    ldi    temp,low(ramend)
    out   spl,temp      ;init sp
    rcall  init_syst    ;init systeme
main:
    sbis   pinc,sa
    rjmp  led_r_off    ;branche si alim ok
led_r_on:
    sbi    portc,ledr
    rjmp  main
led_r_off:
    cbi    portc,ledr
    rjmp  main        ;attn rti

```

Divers.

La fréquence maximale admissible par le AT90S8515-08P est de **8MHz**. Une fréquence de quartz telle que 7.3728MHz peut être utile pour générer les fréquences de communications sérielles sans erreur.

Programmation ISP

La carte MCU8515 permet un mode de programmation ISP (*In Situ Programming*), en liaison avec un ordinateur de développement, un logiciel de dialogue et un câble spécifique branché sur un port LPT.

Le logiciel de liaison a été développé par **JERRY MENG** <http://www.qsl.net/ba1fb/>, il fonctionne sous DOS (ou fenêtre DOS de Windows). Il est très simple d'emploi et dispose de toutes les fonctionnalités nécessaires : lecture et programmation des mémoires internes (FLASH et EEPROM), lecture du code composant, support des circuits AVR 1200, 2313, 2343, 4414, 8515, ...

```
BA1FB AT90SXXXX PC BASED PROGRAMMER V1.60
Copyright 1997,1998,1999 by Jerry Meng
Email:jerrym@public.gb.com.cn

(A) Load HEX file to Flash buffer
(B) Load HEX file to EEPROM buffer
(C) Display Flash buffer
(D) Display EEPROM buffer
(E) Program
(F) Read Device code
(G) Read Flash & EEPROM to buffer
(H) Save Flash buffer to HEX file
(I) Save EEPROM buffer to HEX file
(J) Generate AVR Studio DEBUG.OBJ
(K) Setup
(L) Show schematic
(M) Toggle Reset to high(Current low)
(X) Quit
```

Copie de l'écran principal

```
(1) Erase Flash & EEPROM memory ----- Yes
(2) Program Flash memory ----- Yes
(3) Program EEPROM memory ----- No
(4) Verify Flash memory ----- No
(5) Verify EEPROM memory ----- No
(6) Lock protect bit 1 ----- No
(7) Lock protect bit 2 ----- No
(8) Printer port ----- 1
(9) Device ----- AT90S8515/8535
(0) RC enable (2323/2343 only) ----- No
(S) Save setup
```

Copie de l'écran de configuration

La liaison entre la carte MCU8515 et un ordinateur de développement s'effectue par un port parallèle. Le câble est fourni avec la carte, le brochage est donné dans le menu '*Show schematic*' du logiciel **FBPRG16**.

J.L.S. INFORMATIQUE

2, rue Clément Ader

B.P. 50065

57972 Yutz, Cedex

France

Tél.: 03 82 86 00 16

Fax : 03 82 86 00 12

www.jls-info.com